

## REMARKS/ARGUMENTS

### 1. Claims 1-3, 10-12, and 19-21 are Patentable Over the Cited Art

The Examiner rejected claims 1-3, 10-12, and 19-21 as anticipated by Helgeson (U.S. Patent No. 6,643,652). Applicants traverse with respect to the amended claims.

Amended claims 1, 10, and 19 concern an extensible file access method for accessing a foreign file system from a data processing system with a native file system, said foreign file system and said native file system implementing different file system protocols, wherein files created by the foreign file system may not be accessed using the native file system protocol. The claims require: issuing a request according to a native file system protocol to perform a file system operation with respect to the foreign file system; translating the native file system request to an intermediate programming interface, wherein the intermediate programming interface is different from both the native file system protocol and a foreign file system protocol, and wherein the intermediate programming interface comprises a set of generic access functions common to the native file system protocol and the foreign file system protocol; translating the intermediate programming interface request to a file system request according to the foreign file system protocol; using the translated file system request according to the foreign file system protocol to perform the requested file system operation with respect to the foreign file system; and returning to the data processing system a response from the foreign file system responsive to the translated request in the foreign file system protocol.

Applicants amended claims 1, 10, and 19 to add the requirements that files created by the foreign file system may not be accessed using the native file system protocol (disclosed on at least pg. pg. 4, lines 16-19 of the Application); that the native file system request is to perform a file system operation with respect to a file (disclosed on at least pgs. 17-23 of Application); that the intermediate programming interface comprises a set of generic access functions common to the native file system protocol and the foreign file system protocol (disclosed on at least pg. 15, lines 7-10 of the Application); and using the translated file system request according to the foreign file system protocol to perform the requested read or write operation with respect to the foreign file system (dislcosed on at least pg. 10, lines 5-17; pg. 15, lines 15-21; and pgs. 17-23 which show file system operations that may be performed).

The Examiner cited col. 2, lines 56-67 and col. 11, lines 40-67 as disclosing the “issuing a request” limitation. (Final Office Action, pg. 3) Applicant traverses with respect to the amended form of this limitations which now requires issuing a request according to a native file system protocol to perform a file system operation with respect to a file stored in the foreign file system.

The cited col. 2 discusses how a data object received in a first system local format is translated to a generic interface exchange format using a native API of the first system. The data object is translated from the generic interchange format to a second system specific local format using an API of the second system.

Although the cited col. 2 discusses how to translate a data object from a first to second local fomrat. Nowhere does the cited col. 2 anywhere disclose issuing a request according to a native file system protocol to perform a file system operation with respect to a foreign file system. The cited col. 2 nowhere discloses issuing a request according to a native file system to perform with respect to foreign file system. Instead, the cited col. 2 translates a data object from one format to another.

The cited col. 11 discusses a business applications platform having various components that uses the XML protocol for translating XML into the particular API required by the server and for translating a server’s internal language into XML prior to transmission. These servers may be contained in a single platform and may communicate with each other directly without the necessity of changing the interfacing protocol format.

Although the cited col. 11 discusses how two components, or servers, may communicate by translating to XML, nowhere does the cited col. 11 anywhere disclose issuing a request according to a native file system protocol to perform a file system operation with respect to a file in the foreign file system. Nowhere is there any mention of one server issuing a file system request in a native file system protocol with respect to a foreign file system. Instead, the cited col. 11 discusses using XML to allow two different servers to communicate.

Moreover, nowhere do the cite cols. 2 and 11 anywhere disclose the claim requirement that files created by the foreign file system may not be accessed using the native file system protocol.

The Examiner cited col. 2, lines 58-61 as disclosing the claim requirement of translating the native file system request to the intermediate programming interface. (Final Office Action, pg. 3) Applicants traverse.

The cited col. 2 mentions translating a data object from a first system local format to a generic interchange format. Although the cited col. 2 discusses translating a data object to an intermediate format, nowhere does the cited col. 2 anywhere disclose translating a native file system request to perform an operation with respect to a file in a foreign file system to an intermediate programming interface. Instead, the translation for the cited col. 2 concerns a data object from one format to another.

Applicants amended the translating limitation to further require that the intermediate programming interface comprises a set of generic access functions common to the native file system protocol and the foreign file system protocol. This requirement was added from claims 4, 13, and 22. The Examiner found that col. 2, lines 52-55 of Hegelson teaches this requirement from claims 4, 13, and 22 added to claims 1, 10, and 19. (Final Office Action, pg. 5) Applicants traverse.

The cited col. 2 mentions a plurality of predefined stylesheets each describing a mapping between a local format and the generic interchange format. Nowhere does the cited col. 2 anywhere disclose an intermediate programming interface comprising a set of generic access functions common to the native and foreign file system protocols. Instead, the cited col. 2 discusses stylesheets describing how to map an object from one format to another. Applicants submit that the cited stylesheets defining how to map one object from another does not disclose and is different than the claimed set of generic access functions common to two (native and foreign) file system protocols.

The Examiner cited col. 2, lines 66-67 of Helgeson as disclosing the claim requirement of translating the intermediate programming interface request to a file system request according to the file system request in the foreign file system protocol. (Final Office Action, pg. 3) Applicants traverse.

The cited col. 2 mentions that a native API of the second system is used to translate the data object from the interchange format and the translated data object is transferred to the second system. Again, the cited col. 2 concerns translating an object to a second format. Nowhere does the cited col. 2 anywhere disclose translating an intermediate API to a file system request in a

foreign file system protocol. Applicants submit that a translated data object does not disclose and is different from the claimed translated file system request to the foreign file system protocol.

The Examiner has not cited any art that discloses the added claim requirement of using the translated file system request according to the foreign file system protocol to perform the requested file system operation with respect to the foreign file system. Nowhere does the cited col. 2 anywhere disclose using a translated file system request to perform a file system operation on a foreign file system.

The Examiner cited col. 3, lines 20-21 as disclosing the claim requirement of returning to the data processing system a response from the foreign file system responsive to the translated request in the foreign file system protocol. (Final Office Action, pg. 3) Applicants traverse.

The cited col. 3 mentions that a first component comprises a system independent service subcomponent that utilizes a native API of the first system to translate the data object to the generic interchange format object. Thus, the cited col. 3 concerns translating the object in the first system to the generic format object. Nowhere does this cited col. 3 anywhere disclose returning to the data processing system issuing the native file system request a response from the foreign file system responsive to the translated request. Instead, the cited col. 3 discusses how a first component translates a data object to a generic format object using native APIs. There is no disclosure or mention of returning data response to a translated file system request as claimed.

Accordingly, claims 1, 10, and 19 are patentable over the cited art because the cited Hegelson does not disclose the requirements of these claims.

Claims 2, 3, 11-12, 20, and 21 are patentable over the cited art because they depend from claims 1, 10, and 19, which are patentable over the cited art for the reasons discussed above. Moreover, the following discussed dependent claims provide additional grounds of patentability over the cited art.

Claims 2, 11, and 20 depend from claims 1, 10, and 19 and further require that the file access method is extended to support a second foreign file system by providing a translation from the intermediate programming interface to the second foreign file system protocol, said method further comprising determining the foreign file system protocol.

The Examiner cited col. 2, lines 61-66, col. 56, lines 57-67, and col. 57, lines 1-67 as teaching the additional requirements of these claims. (Final Office Action, pgs. 3-4) Applicants traverse.

The cited col. 2 mentions that a data object is translated from a generic interchange format to a second system local format object with predefined stylesheets. Nowhere does the cited col. 2 anywhere disclose that a file request may be translated to a first or second foreign file system and determining the foreign file system protocol. There is no mention or disclosure in the cited col. 2 of translating a native file system request to one of two foreign file system requests. Instead, the cited col. 2 discusses translating a data object from the generic format to the second system local format.

The cited col. 56 mentions that when a client sends a request to a web content server application the control file is identified and parsed. Nowhere does the cited col. 56 anywhere disclose that a file request may be translated to a first or second foreign file system and determining the foreign file system protocol.

The cited col. 57 mentions that the control file contains a processing instruction that is used to look-up a processor to process a document. The control file processor performs various operations, such as identifying model, view and widget files, parsing the model to create an XML DOM, etc. Additional operations are discussed. The Examiner has not cited any part of col. 57 that discloses that a file request may be translated to a first or second foreign file system and determining the foreign file system protocol.

Accordingly, claims 2, 11, and 20 provide additional grounds of patentability over the cited art because the additional requirements of these claims are not disclosed in the cited art.

2. Claims 4-9, 13-18, and 22-27 are Patentable Over the Cited Art

The Examiner rejected claims 4-9, 13-18, and 22-27 as obvious (35 U.S.C. §103) over Helgeson in view of Bodamer. Applicants traverse.

First off, claims 4-9, 13-18, and 22-27 are patentable over the cited art because they depend from base claims 1, 10, and 19, which are patentable over the cited art for the reasons discussed above. Moreover, the following dependent claims provide further grounds of patentability over the cited art.

Applicants amended claims 4, 13, and 22 to remove the limitation added to base claims 1, 10, and 19. These amended claims require that the intermediate programming interface further comprises a set of file system specific functions which are not common to the file system protocols. The Examiner cited col. 3, lines 6-10, 6, lines 21-20, col. 17, lines 57-67, and col. 18, lines 1-2 of Bodamer as teaching these requirements of claims 4, 13, and 22. (Final Office Action, pg. 5) Applicants traverse.

The cited col. 3 discusses a local database server having a heterogeneous services module to dispatch a request to an agent process in communication with a foreign database server. The foreign database server executes the statement and the heterogeneous service module sends the request to the agent process, which sends the request to the foreign database server. Nowhere does this cited col. 3 anywhere teach or suggest that an intermediate programming interface, that translates file system operations between the native and foreign file system protocols, include a set of file system specific functions which are not common to the file system protocols. The cited col. 3 discusses how requests are sent between a local database server and a foreign database server.

The cited col. 6 mentions that the heterogeneous service module identifies an operation necessary for execution of the client statement that must be executed by an external process, such as the foreign database server and forwards the operation to the appropriate agent. Nowhere does this cited col. 6 anywhere teach or suggest that intermediate programming interfaces, that translates file system operations between the native and foreign file system protocols, include a set of file system specific functions which are not common to the file system protocols. Instead, the cited col. 6 discusses how a statement is forwarded to an agent.

The cited col. 17 mentions if the DDX 231 does not include a reference to a specified table, then the local server 202 checks to see if the user's schema defines the unknown table and if not, the local server 202 checks its internal data dictionary generally. The DDX 231 includes table definitions that are perceived to be in the foreign database system 208, even though the foreign system 208 may not have the specified table, or the foreign database system may not store the requested information in a compatible database structure. For example, the foreign database may not support object definitions using data dictionary tables. In this case, the DDX 231 table specifies that the unknown table is a mimicked table.

The DDX is a dictionary translation table. If the local server receives a query from a client 208 referencing a table in a foreign database system 208, then the local server 202 first attempts to use that table on the foreign database system 208. If the table does not exist at the foreign database system 208, an error is returned and, in response, the heterogeneous services module 311 checks a data dictionary translation table (DDX) 231. If translation information is found, the query is translated and sent to the foreign database system 208 via the agent 300. The results of the query from the foreign database system 208 are then translated before being back sent to the client 200. (Badamer, col. 9, lines 1-14)

The cited DDX is a dictionary translation table that is checked for a translation of a query referencing a table in the foreign database systems. Nowhere does this cited col. 17 anywhere teach or suggest that intermediate programming interfaces, which translates file system operations between the native and foreign file system protocols, include a set of file system specific functions which are not common to the file system protocols. Instead, the DDX translates references to a foreign database table.

The cited col. 18 mentions that the DDX specifies whether an unknown table is a mimicked table. Although the cited col. 18 discusses a dictionary specifying whether a table is an unknown table, nowhere does this cited col. 17 anywhere teach or suggest that intermediate programming interfaces, which translate file system operations between the native and foreign file system protocols, include a set of file system specific functions which are not common to the file system protocols.

Accordingly, claims 4, 13, and 23 provide additional grounds of patentability over the cited art because the additional requirements of these claims are not disclosed in the cited art.

Claims 5, 14, and 24 depend from claims 4, 13, and 23 and further require that the set of generic access functions common to the native file system protocol and the foreign file system protocol are translated from the native file system protocol to the intermediate programming interface which is then translated to the foreign file system protocol, and wherein the set of file system specific functions which are not common to the file system protocols are not translated from the native file system protocol to the intermediate programming interface which is then translated to the foreign file system protocol.

The Examiner cited col. 18, lines 3-11 of Bodamer as teaching the claim requirement that the set of file system specific functions which are not common to the file system protocols are not translated from the native file system protocol to the intermediate programming interface which is then translated to the foreign file system protocol. (Final Office Action, pgs 5-6) Applicants traverse.

The cited col. 18 mentions that if the foreign database does not support definitions using dictionary tables, then dictionary table DDX 231 specifies that the unknown table is a mimicked table. Nowhere does this cited col. 18 anywhere teach or suggest that the set of file system specific functions which are not common to the file system protocols are not translated from the native file system protocol to the intermediate programming interface.

Accordingly, claims 5, 14, and 24 provide additional grounds of patentability over the cited art because the additional requirements of these claims are not disclosed in the cited art.

Claims 6-9, 15-18, and 25-27 are additionally patentable over the cited art because they depend from claims 5, 14, and 24, which provide additional grounds of patentability, and because the additional requirements of these claims in combination with the base claims provide further grounds of patentability over the cited art.

#### 1. New Claims 28

New claims 28, 31, and 34 depend from claims 1, 10, and 19 and further require that the request to perform the file system operation with respect to the foreign file system comprises a request to create or delete a directory in the foreign file system. The additional requirements of these claims are disclosed on at least pages 17 and 22 of the Application.

These added claims 28, 31, and 34 are patentable over the cited art because the base claims from which they depend, claims 1, 10, and 19, respectively, are patentable over the cited art and because the combination of these new claim requirements with the base claims provides additional grounds of patentability over the cited art.

New claims 29, 32, and 35 depend from claims 1, 10, and 19 and further require that the request to perform the file system operation with respect to the foreign file system comprises a request to delete a file in the foreign file system. The additional requirements of these claims are disclosed on at least page 17 of the Application.

These added claims 29, 32, and 35 are patentable over the cited art because the base claims from which they depend, claims 1, 10, and 19, respectively, are patentable over the cited art and because the combination of these new claim requirements with the base claims provides additional grounds of patentability over the cited art.

New claims 30, 33, and 36 depend from claims 1, 10, and 19 and further require that the request to perform the file system operation with respect to the foreign file system comprises a request to rename a file in the foreign file system. The additional requirements of these claims are disclosed on at least page 22 of the Application.

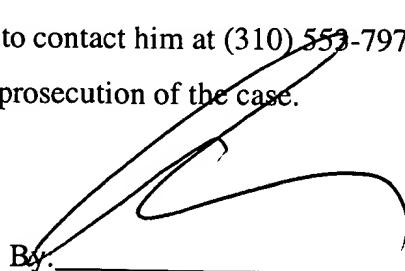
These added claims 29, 32, and 35 are patentable over the cited art because the base claims from which they depend, claims 1, 10, and 19, respectively, are patentable over the cited art and because the combination of these new claim requirements with the base claims provides additional grounds of patentability over the cited art.

### Conclusion

For all the above reasons, Applicant submits that the pending claims 1-35 are patentable over the art of record. Applicants submit herewith the fee for the added claims and extension of time. Nonetheless, should any additional fees be required, please charge Deposit Account No. 09-0460.

The attorney of record invites the Examiner to contact him at (310) 553-7977 if the Examiner believes such contact would advance the prosecution of the case.

Dated: October 27, 2004

By: 

David W. Victor  
Registration No. 39,867

Please direct all correspondences to:

David Victor  
Konrad Raynes & Victor, LLP  
315 South Beverly Drive, Ste. 210  
Beverly Hills, CA 90212  
Tel: 310-553-7977  
Fax: 310-556-7984